

EV316936 340

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Pane Element

Inventor:

Brian Tunning

ATTORNEY'S DOCKET NO. MS1-1484US

CROSS-REFERENCE TO RELATED APPLICATIONS

The instant application is related to co-pending U.S. Patent Application Ser. No. _____, Applicant Docket No, MS1-1483US, entitled "Content Template System" by Brian Tunning which is filed concurrently herewith, assigned to the assignee of the present invention, and incorporated herein by reference for all that it teaches and discloses.

TECHNICAL FIELD

The subject matter relates generally to web page authoring and more specifically to an exemplary pane element.

BACKGROUND

Software applications for standalone computing devices often feature sophisticated and interactive graphical user interfaces (GUIs) having windows. Dialogue windows for user input (i.e., modal windows) can be dazzling in the interactive detail of their text dialogue boxes, buttons, and icons and in the power and variety of editing tools made available to a user. Even non-modal windows are highly customized in many software applications and sophisticated in the manner in which they change focus, layer with each other, and interact with a primary window. Pop-up menus, having layered submenus and shortcut keys are taken for granted. All these elegant GUIs, however, are difficult to reproduce in web pages to be used on an internet because web pages generally use web content publishing languages, such as hypertext markup language (HTML). Application programmers and web page authors have very different tools at their disposal.

1 Conventionally, markup languages, such as HTML, do not have the
2 interface capability of a GUI platform for a software application. This increases
3 the skill required for entry-level and even experienced web designers to produce
4 web pages that have sophisticated interactive functionality: the web designer must
5 also be a programmer, or at least rely on one.

6 Dynamic HTML (DHTML) and web scripting have increased the capability
7 of HTML and other languages that are popular for creating web pages to perform
8 with more sophisticated interactivity. But this improvement still does not
9 approach that of software application GUIs, especially with respect to web pages
10 used with different types of browsers. A web page author contemplating either a
11 sophisticated GUI or a customized window in a web page has limited choices:
12 write additional web pages to create entirely separate windows (serious
13 programming commitment), or use "canned" web page code to make stock
14 windows (not very customizable). Writing another web page, to be created as an
15 entirely separate window, e.g., for an interactive dialogue, complicates accessing
16 variables and/or content on the primary web page originating the new web page
17 window and requires the web page developer to write event-trapping code to
18 invoke the dialogue.

19 **Fig. 1** shows a conventional website 100 appearing on a display screen 102
20 of a computing device 104, wherein a web page document 106 includes a program
21 listing having markup instructions 108, 110 that are rendered by a web browser
22 112 into a graphical user interface (GUI) consisting of multiple web page windows
23 114, 116. (A web page document 106 is to be distinguished from a content
24 document, such as a text document, that is displayed as textual content in a
25 rendered web page.) An introductory or home web page window 114 typically

1 links or branches to a more interactive web page window 116 having one or more
2 interactive user interfaces (UIs) 118, 120, 122 if a user decides to engage in a
3 dialogue or transaction via input and/or editing controls of the interactive web
4 page window 116. To present an interactive web page window 116 having
5 interactive UIs 118, 120, 122 suitable for substantial dialogue or transactions, a
6 web designer typically embarks on multiple separate web page projects, one for
7 each web page window associated with the website 100. Having to design a
8 separate web page for each window may discourage some web designers,
9 especially when some of the windows to be used in a website 100 include
10 relatively complex interactive dialogue and/or transaction controls.

11 12 **SUMMARY**

13 Subject matter includes an exemplary pane element for use in a web page
14 document to generate a pane consisting of a division of a computer display screen.
15 When a web browser encounters the exemplary pane element, the browser is
16 directed to pane-generating instructions, which are substituted as markup language
17 for the exemplary pane element. The pane-generating instructions may be an
18 element behavior bound to an exemplary pane element.

19 20 **BRIEF DESCRIPTION OF THE DRAWINGS**

21 Fig. 1 is a graphic representation of a conventional method of generating
22 web page windows.

23 Fig. 2 is a graphic representation of generating an exemplary web page
24 pane using an exemplary pane element.
25

1 Fig. 3 is a graphic representation of exemplary syntax for an exemplary
2 pane element.

3 Fig. 4 is a graphic representation of another exemplary syntax for an
4 exemplary pane element.

5 Fig. 5 is a graphic representation of exemplary logic associated with an
6 exemplary pane element.

7 Fig. 6 is a graphic representation of multiple exemplary pane elements
8 generating multiple exemplary web page panes.

9 Fig. 7 is a graphic representation of an exemplary interaction between
10 exemplary web page panes as coordinated by exemplary underlying pane-
11 generating logic.

12 Fig. 8 is a graphic representation of another exemplary interaction between
13 exemplary cascaded web page panes as coordinated by exemplary underlying
14 pane-generating logic.

15 Fig. 9 is a graphic representation of an exemplary interaction generated by
16 exemplary pane-generating logic between two exemplary web page panes, based
17 on changing content in one of the exemplary web page panes.

18 Fig. 10 is a block diagram of an exemplary pane engine.

19 Fig. 11 is a flow diagram of an exemplary method of generating an
20 exemplary web page pane using an exemplary pane element.

21 Fig. 12 is a block diagram of an exemplary computing device suitable for
22 use with the subject matter.

23

24

25

DETAILED DESCRIPTION

Overview

Subject matter presented herein includes an exemplary markup language element referred to as an exemplary “pane” element or tag. When rendering an exemplary pane element, a web browser is referred to pane-generating instructions underlying the exemplary pane element. The pane-generating instructions may be an element behavior bound to the exemplary pane element. The pane-generating instructions may be substituted in markup language form for the exemplary pane element during execution by a web browser. This allows sophisticated, custom pane-generating logic to be used in a web page-producing document, such as an HTML document, without the web designer having to be proficient in programming the pane-generating instructions.

An exemplary pane is a division of a computer display screen, usually a rectangular area that has boundaries. Exemplary panes can be opened, closed, and moved around on a computer display screen and may contain interactive elements of a graphical user interface (GUI). According to one aspect of the subject matter, a user can adjust dimensions (size and shape), format, color, and placement, etc., of exemplary panes, and the exemplary panes can be overlapped (i.e., layered, cascaded) to increase the usable display screen area of a web page. The overlapping panes can be brought to the front or sent to the back of a group of exemplary panes perceived as a 3-dimensional stack of panes on 2-dimensional display screen surface. The active (“in focus”) exemplary pane is that which a user is currently using, and typically appears in front when exemplary panes are overlapped. Exemplary panes may be scrollable to reveal, for example, different

1 areas of a large pane. Different content, applications, or files can be presented on
2 a display screen simultaneously in different exemplary panes. Of course, more
3 than one exemplary pane can be open simultaneously and more than one can be
4 partly or fully visible at once.

5 An exemplary pane element can encapsulate static and/or dynamic markup
6 language content, allowing a user to customize the layout of a web page for
7 specific needs, such as screen resolution, irregularity of content shape, and priority.
8 An exemplary pane element significantly lowers the entry-level skill needed for
9 web page designers to create more interactive and customizable user interfaces
10 (UIs).

11 **Fig. 2** shows an exemplary website 200 presented on a display screen 102
12 of a computing device 104. The exemplary website 200 includes a web page 202
13 generated by a program listing having exemplary markup instructions 204. The
14 markup instructions 204 are written in a “web page language,” such as HTML or
15 another markup language, such as extensible markup language (XML).
16 Languages used to generate a web page 202 are usually limited to those that a web
17 browser can interpret. In this case, the web page document is an exemplary
18 HTML document 206. The exemplary HTML document 206 generates a web
19 page 202 when rendered by a web browser 112.

20 In one implementation of the subject matter, the exemplary HTML
21 document 206 includes an exemplary pane element 208, which generates and/or
22 designates an exemplary pane 210. To describe an exemplary pane 210,
23 comparison with a conventional web page “window” is helpful. Conventionally,
24 each window appearing on a display screen 102 is roughly equated with separate
25 markup instructions (e.g., 108, 110) for generating separate web page windows

1 (e.g., 114, 116). Instances of an exemplary pane element 208, by contrast, can
2 generate one or many exemplary panes 210 from within markup instructions 204
3 for a single web page 202 without embarking on programming for separate web
4 pages. Thus, since an exemplary pane element 208 typically automates much of
5 the work of developing separate web page windows, the exemplary pane element
6 208 can be employed by a novice web designer to conveniently generate one or
7 more exemplary panes 210 from within a unified set of markup instructions 204
8 without designing each web page window manually.

9 The amount of automation underlying a particular implementation of an
10 exemplary pane element 208 is not limited, as the underlying pane-generating
11 instructions bound to an implementation of an exemplary pane element 208 can be
12 lengthy and/or sophisticated.

13 In one implementation of the subject matter, an exemplary pane element
14 208 may be used in an HTML document 206 in the following general format:

15 <pane>
16 ← insert content of pane here
17 </pane>

18 This example format demonstrates that content to be displayed in the
19 generated exemplary pane 210 can be inserted between an opening markup
20 indicator "tag" and a closing tag.

21 The actual size of a pane 210 generated and/or designated by an exemplary
22 pane element 208 may depend on the web browser 112 that is employed to render
23 the HTML document 206. Although web browsers may differ in how they
24 interpret and display a web page 202, the interpretive characteristics of a given
25 browser are generally consistent so that the size of an exemplary pane 210 remains

1 in proportion to other panes, e.g., a primary or “home” web page or other windows
2 and/or panes generated by other instances of the exemplary pane element 208
3 within an HTML document 206.

4 **Fig. 3** shows an exemplary syntax 300 of an exemplary pane element 208.
5 In an exemplary opening pane tag 302, implementations of the subject matter can
6 include one or more pane attributes 304 (and associated values). The pane
7 attributes 304 allow a web designer to optionally specify initial dimensions,
8 placement within a web page, etc., of an exemplary pane 210. Thus, an exemplary
9 opening pane tag 302 may have multiple pane attributes 304 by which
10 implementation details of a particular exemplary pane 210 can be abstracted from
11 a web designer via selection of attribute values. A prefix 306, such as system
12 name, may also be used. A closing tag 308 ends the exemplary pane element 208,
13 closing in content 310 to be displayed in an exemplary pane 210.

14 Alternatively, as shown in **Fig. 4**, an exemplary pane element 208 can have
15 few or no attributes. This type of implementation may be desirable when an
16 exemplary pane 210 that has consistent characteristics and/or content is to be used
17 repeatedly in a web page design. In one implementation, an exemplary pane
18 element 208 has only an opening pane tag 302 and no closing tag 308.

19 As shown in **Fig. 5**, when a web browser 112 renders an exemplary pane
20 element 208, the web browser 112 is referred to a body of associated logic 500
21 (i.e., pane-generating instructions) underlying the exemplary pane element 208. If
22 associated logic 500 underlying an exemplary pane element 208 is well developed,
23 then an exemplary pane element 208 that has few or no manually selectable
24 attributes may be able to automatically determine pane attributes. On the other
25 hand, well-developed associated logic 500 may also allow a web designer to

1 manually select more kinds of attributes for describing and selecting an exemplary
2 pane 210 to generate than would be possible with rudimentary associated logic
3 500.

4 **Fig. 6** shows exemplary pane elements 600, 602 that have associated logic
5 500 to dynamically adjust the dimensions of resulting exemplary panes 604, 606
6 based on the amount of content to be displayed in each exemplary pane 604, 606.
7 Exemplary self-adjusting can be applied to many other attributes of exemplary
8 panes 604, 606, not just to dimension attributes. For example, attributes of an
9 exemplary pane (e.g., 210) may be based on details of user interface controls (e.g.,
10 118, 120, 122) to be included in the exemplary pane 210. There are many
11 characteristics of an exemplary pane 604, 606 that can be determined
12 automatically before the exemplary pane 210 is generated on a display screen 102.
13 In fact, an exemplary pane element 602 can encompass content 601 having content
14 attributes, such as size, shape, color, format, style, and font and these content
15 attributes can also be used to automatically determine pane attributes 304 (for
16 instance, pane dimensions). For example, a large headline font may result in an
17 increased height dimension for an exemplary pane 606.

18 In some implementations the associated logic 500 discussed above can take
19 the form of an exemplary pane engine 1000 (to be discussed more fully below)
20 that can determine characteristics of content 601 associated with an exemplary
21 pane element 602 and automatically generate (attributes for) an exemplary pane
22 606 having characteristics needed to display the content 601. For example, an
23 exemplary pane engine 1000 may determine amounts and attributes of header
24 content, text content, link resources, interactive user interfaces, etc., and create an
25

1 exemplary pane 606 that has appropriate dimensions, color, font, format,
2 placement, etc.

3 **Fig. 7** shows an exemplary interaction 700 between two exemplary panes
4 702, 704 displayed simultaneously. Associated logic 500 bound to exemplary
5 pane elements 706, 708 coordinates interrelated changes in the two exemplary
6 panes 702, 704. In a scenario in which panes are tiled instead of cascaded, when a
7 first exemplary pane 702 becomes narrower, a second exemplary pane 704
8 becomes wider to make use of the newly freed space. Thus, attributes of a given
9 exemplary pane 704 can adapt to changes in attributes of another exemplary pane
10 702.

11 An exemplary pane 702 may have its size, position, color, z-order priority,
12 style, etc., changed because of corresponding or non-corresponding changes in
13 another exemplary pane 704 being simultaneously displayed or due to the advent
14 of a new exemplary pane. If a new exemplary pane 704 is generated, other
15 exemplary panes 702 may be faded to a less focused color and placed visually
16 behind the new exemplary pane 704 in a perceived stack of panes.

17 **Fig. 8** shows an exemplary interaction 800 between multiple exemplary
18 panes 802, 804, 806 displayed simultaneously and cascaded based on z-order
19 priority. When a web browser 112 encounters an exemplary pane element 600, a
20 new exemplary pane 802 generated by the exemplary pane element 600 is
21 typically placed by associated logic 500 of the exemplary pane element 600 on top
22 of a perceived 3-dimensional cascaded stack of pre-existing exemplary panes 802,
23 804, 806 appearing on the 2-dimensional surface of a display screen 102.

24 The focus and/or z-order priority of an exemplary pane 804 in the stack
25 may change if a user summons the exemplary pane 804 from the stack by clicking

1 on it with a mouse, etc. The associated logic 500 of the exemplary pane element
2 600 can orchestrate the z-order of multiple exemplary panes 802, 804, 806 in a
3 cascaded stack of panes, or can rearrange a placement order of the exemplary
4 panes 802, 804, 806 when they are arranged in tile format (no cascading).

5 **Fig. 9** shows exemplary interactions 900 between two exemplary panes
6 902, 904 that are simultaneously, based on changing content. Changes, such as
7 edits, to content 901 in a first exemplary pane 902 are simultaneously displayed in
8 a second exemplary pane 904. These exemplary interactions 900 are controlled by
9 associated logic 500 of the exemplary pane elements 600, 602, which generate the
10 two exemplary panes 902, 904. If a first exemplary pane 902 displays content and
11 the content is changed, then a second exemplary pane 904 may have not only its
12 content changed to reflect the change in the other exemplary pane 904, but may
13 have its pane attributes changed as well. A dynamic change in content reflected in
14 both exemplary panes 902, 904 may be provided by exemplary edit, preview, and
15 resource tags usable in an HTML document 206, as described in co-pending U.S.
16 Patent Application Ser. No. _____, entitled "Content Template System"
17 by Brian Tunning. When a first exemplary pane 902 displays content and editing
18 controls to change the content, the second exemplary pane 904 displays the edits
19 to the content in real time. Thus, a real-time web page preview of edits made in an
20 HTML document 206 is accomplished simultaneously with the editing using
21 exemplary pane elements 902, 904. The real-time link between dynamically
22 changing content associated with one exemplary pane 902 and a simultaneous
23 version of the same dynamically changing content displayed in another exemplary
24 pane 904 can be accomplished via exemplary associated logic 500, wherein the
25 content 901 being edited in the first exemplary pane 902 becomes a content

1 resource for the real-time web page being displayed in the second exemplary pane
2 904.

3 Implementations Using Element Behaviors

4 In some implementations, the pane-generating instructions (i.e., associated
5 logic 500 of the exemplary pane element 208) are achieved using a bound, custom
6 “element behavior.” When a browser, such as versions of MICROSOFT®
7 INTERNET EXPLORER (“IEExplorer”), encounters an exemplary pane element
8 208, additional element behavior code is substituted for the exemplary pane
9 element 208 and executed to provide the functionality of the exemplary pane
10 element 208. The additional code of an element behavior is usually performed on
11 the client side.

12 An element behavior may be expressed as underlying associated logic 500,
13 an exemplary pane engine 1000 (to be discussed below), and/or other pane-
14 generating instructions, which then become usable via an exemplary pane element
15 208 in the same way as “everyday” HTML elements in a web page HTML
16 document 206. Element behaviors can be written in script using an HTML
17 component (HTC) file, or implemented with a DHTML behavior. An element
18 behavior can then be downloaded, parsed, and imported into web page execution
19 via a special processing instruction and permanently bound to an exemplary pane
20 element 208.

21 Browsers that enable HTC files to define an exemplary pane element 208
22 provide the quickest and easiest way to create element behaviors using scripting
23 languages such as MICROSOFT VISUAL BASIC® SCRIPTING EDITION
24 (VBSCRIPT), MICROSOFT JSCRIPT®, MICROSOFT WINDOWS® SCRIPT
25 COMPONENT (WSC), and/or MICROSOFT VISUAL C++®. HTC files provide a

1 mechanism to implement components in script as DHTML behaviors. Saved with
2 an “.htc” extension, an HTC is an HTML file that contains script and a set of HTC-
3 specific elements that define the exemplary pane element 208.

4 Whereas a kind of behavior called an “attached behavior” (bound
5 asynchronously to an HTML element) overwrites the default behavior of an
6 element to which it is attached, element behaviors define new elements, such as an
7 exemplary pane element 208. A web page document that employs exemplary
8 pane elements 208 typically has improved organization and less clutter, since
9 element behaviors are encapsulated in an HTC file. DHTML behaviors defined
10 with HTC files can be used in web pages within the same domain, so element
11 behaviors are an excellent way of creating powerful, reusable exemplary pane
12 elements 208. Once an element behavior has been parsed and initialized, it
13 renders an exemplary pane element 208 into a genuine HTML element, as the
14 element behavior cannot be disconnected from the exemplary pane element 208.

15 To create and use an exemplary pane element 208 in a web page document
16 the element behavior of the exemplary pane element 208 is first defined in an HTC
17 file and then the definition is imported into the web page document.

18 In some implementations, an HTML document 206 requires a few
19 additional markings to import a definition of an element behavior of an exemplary
20 pane element 208. For example, an HTML document 206 first declares a
21 namespace, which is used to ensure that the exemplary pane element 208 has a
22 unique qualifier. Then the associated logic 500 of an exemplary pane element 208
23 is imported into the namespace. Thus, a “pane” namespace imports the
24 implementation of an exemplary pane element 208 from a pane.htc file:
25

1 <?IMPORT namespace="pane" implementation="pane.htc" >

2 In some implementations, the import directive can be important in
3 implementing an element behavior in the web page document. When a web
4 browser 112 begins processing the import directive, it waits until the contents of
5 the corresponding HTC file have downloaded completely before continuing. The
6 manner of processing then causes the element behavior to be bound synchronously
7 to the exemplary pane element 208.

8 The exemplary pane element 208 itself must be inserted somewhere in the
9 body of the web page document. An exemplary pane element 208 can be used like
10 any other tag, except that the exemplary pane element 208 may use its namespace
11 as the prefix.

12 Exemplary Pane Engine

13 **Fig. 10** shows an exemplary pane engine 1000, which may be invoked or
14 referenced as a form of associated logic 500 and/or as an element behavior when a
15 web browser 112 encounters an exemplary pane element 208. That is, an
16 exemplary pane engine 1000 is one way to represent the pane-generating
17 instructions and/or the underlying associated logic 500 discussed above that is
18 associated with an exemplary pane element 208, for example, by being bound to
19 the exemplary pane element 208 as an element behavior.
20

21 An exemplary pane engine 1000 may contain components to execute an
22 exemplary pane 210 (or assist a web browser 112 to render the exemplary pane
23 210). A pane sizer 1002 establishes dimensions of an exemplary pane 210 to be
24 generated, and a pane positioner 1004 determines a placement of the exemplary
25 pane 210 on a display screen 102. These attributes (pane dimensions and position)

1 for executing an exemplary pane 210 may be obtained from a pane attribute
2 assignor 1006. In one implementation, a pane attribute assignor 1006 may include
3 default pane attributes 1010, so that when an exemplary pane element 600 is
4 encountered, a working set of pane attributes 304 is immediately available. An
5 attribute reader 1008 may also be included in the pane attribute assignor 1006 to
6 read attributes supplied by a web designer if the syntax of an exemplary pane
7 element 208 allows attributes to be manually specified in a web page document.

8 In some implementations, an exemplary pane engine 1000 includes a
9 multipane coordinator 1012, which allows an exemplary pane engine 1000 to
10 consider the attributes and behavior of already-implemented exemplary panes 604
11 when determining attributes to assign to a new exemplary pane 606.

12 If multiple exemplary panes are cascaded (e.g., 802, 804, 806) then a z-
13 order prioritizer 1014 and a focus assignor 1016 may determine the priority (i.e.,
14 depth, layer, order, etc.) of each exemplary pane 802 in the cascaded stack, and
15 also which exemplary pane 802 has focus (i.e., is the top, active exemplary pane
16 802).

17 An attribute interrelation module 1018 analyzes how pane attributes and/or
18 content attributes of a first exemplary pane 606 affect a second exemplary pane
19 606. Accordingly, an attribute interrelation module 1018 includes a pane attribute
20 analyzer 1020 and a content attribute analyzer 1022. The pane attributes of a one
21 exemplary pane 606 may be affected by changes in pane attributes and/or content
22 attributes of another exemplary pane 604. For example, in some implementations
23 the amount of content displayed in a first exemplary pane 604 might affect the size
24 of a second exemplary pane 606. The attribute interrelation module 1018 analyzes
25

1 whether pane attributes and content attributes of already-existing exemplary panes
2 604 should affect a given exemplary pane 606 to be generated.

3 A dynamic content linker 1024 may be included in the exemplary pane
4 engine 1000 so that changes (e.g., edits) of content 901 displayed in one
5 exemplary pane 902 can be simultaneously displayed in another exemplary pane
6 904.

7 A common resource linker 1026 may also be included in an exemplary
8 pane engine 1000. Two exemplary panes 902, 904 may be linked to a common
9 content resource, for example a file or an input device, such as a keyboard. When
10 content in the common resource changes (e.g., a user actuates keys on the
11 keyboard), the change is reflected in both of the exemplary panes 902, 904.

12 Various other configurations of an exemplary pane engine 1000 are
13 possible. The illustrated implementation is presented only as an example.

14 15 Exemplary Method

16 **Fig. 11** shows an exemplary method 1100 of generating an exemplary pane
17 using an exemplary pane element in a web page document. In the flow diagram,
18 the operations are summarized in individual blocks. The operations of the
19 exemplary method 1100 may be performed in hardware and/or as machine-
20 readable instructions (software or firmware) that can be executed by a processor or
21 a device, such as a device that includes an exemplary pane engine 1000.

22 At block 1102, an exemplary pane element in a web page document is
23 encountered by a web browser.

24 At block 1104, the web browser is referred to pane-generating instructions
25 and/or logic associated with the exemplary pane element.

1 At block 1106, the pane-generating instructions are substituted for the
2 exemplary pane element.

3 At block 1108, the pane-generating instructions are executed to generate an
4 exemplary pane.

5 6 Exemplary Computing Device

7 **Fig. 12** shows an exemplary computer 1200 providing a computing
8 environment in which aspects of the subject matter could be practiced. The
9 components of exemplary computer 1200 may include, but are not limited to, a
10 processing unit 1220, a system memory 1230, and a system bus 1221 that couples
11 various system components including the system memory 1230 to the processing
12 unit 1220. The system bus 1221 may be any of several types of bus structures
13 including a memory bus or memory controller, a peripheral bus, and a local bus
14 using any of a variety of bus architectures. By way of example, and not limitation,
15 such architectures include Industry Standard Architecture (ISA) bus, Micro
16 Channel Architecture (MCA) bus, Enhanced ISA (EISAA) bus, Video Electronics
17 Standards Association (VESA) local bus, and Peripheral Component Interconnect
18 (PCI) bus also known as the Mezzanine bus.

19 Exemplary computer 1200 typically includes a variety of computer-
20 readable media. Computer-readable media can be any available media that can be
21 accessed by exemplary computer 1200 and includes both volatile and nonvolatile
22 media, removable and non-removable media. By way of example, and not
23 limitation, computer-readable media may comprise computer storage media and
24 communication media. Computer storage media include volatile and nonvolatile,
25 removable and non-removable media implemented in any method or technology

1 for storage of information such as computer-readable instructions, data structures,
2 program modules, or other data. Computer storage media includes, but is not
3 limited to, RAM, ROM, EEPROM, flash memory or other memory technology,
4 CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic
5 cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices,
6 or any other medium which can be used to store the desired information and which
7 can be accessed by exemplary computer 1200. Communication media typically
8 embodies computer-readable instructions, data structures, program modules or
9 other data in a modulated data signal such as a carrier wave or other transport
10 mechanism and includes any information delivery media. The term "modulated
11 data signal" means a signal that has one or more of its characteristics set or
12 changed in such a manner as to encode information in the signal. By way of
13 example, and not limitation, communication media includes wired media such as a
14 wired network or direct-wired connection and wireless media such as acoustic,
15 RF, infrared and other wireless media. Combinations of any of the above should
16 also be included within the scope of computer readable media.

17 The system memory 1230 includes computer storage media in the form of
18 volatile and/or nonvolatile memory such as read only memory (ROM) 1231 and
19 random access memory (RAM) 1232. A basic input/output system 1233 (BIOS),
20 containing the basic routines that help to transfer information between elements
21 within exemplary computer 1200, such as during start-up, is typically stored in
22 ROM 1231. RAM 1232 typically contains data and/or program modules that are
23 immediately accessible to and/or presently being operated on by processing unit
24 1220. By way of example, and not limitation, Fig. 12 illustrates operating system
25 1234, the exemplary pane engine 1000, application programs 1235, other program

1 modules 1236, and program data 1237. Although the exemplary pane engine 1000
2 is depicted as software in random access memory 1232, other implementations of
3 an exemplary pane engine 1000 can be hardware or combinations of software and
4 hardware.

5 The exemplary computer 1200 may also include other removable/non-
6 removable, volatile/nonvolatile computer storage media. By way of example only,
7 Fig. 12 illustrates a hard disk drive 1241 that reads from or writes to non-
8 removable, nonvolatile magnetic media, a magnetic disk drive 1251 that reads
9 from or writes to a removable, nonvolatile magnetic disk 1252, and an optical disk
10 drive 1255 that reads from or writes to a removable, nonvolatile optical disk 1256
11 such as a CD ROM or other optical media. Other removable/non-removable,
12 volatile/nonvolatile computer storage media that can be used in the exemplary
13 operating environment include, but are not limited to, magnetic tape cassettes,
14 flash memory cards, digital versatile disks, digital video tape, solid state RAM,
15 solid state ROM, and the like. The hard disk drive 1241 is typically connected to
16 the system bus 1221 through a non-removable memory interface such as interface
17 1240, and magnetic disk drive 1251 and optical disk drive 1255 are typically
18 connected to the system bus 1221 by a removable memory interface such as
19 interface 1250.

20 The drives and their associated computer storage media discussed above
21 and illustrated in Fig. 12 provide storage of computer-readable instructions, data
22 structures, program modules, and other data for exemplary computer 1200. In Fig.
23 12, for example, hard disk drive 1241 is illustrated as storing operating system
24 1244, application programs 1245, other program modules 1246, and program data
25 1247. Note that these components can either be the same as or different from

1 operating system 1234, application programs 1235, other program modules 1236,
2 and program data 1237. Operating system 1244, application programs 1245, other
3 program modules 1246, and program data 1247 are given different numbers here
4 to illustrate that, at a minimum, they are different copies. A user may enter
5 commands and information into the exemplary computer 1200 through input
6 devices such as a keyboard 1262 and pointing device 1261, commonly referred to
7 as a mouse, trackball, or touch pad. Other input devices (not shown) may include
8 a microphone, joystick, game pad, satellite dish, scanner, or the like. These and
9 other input devices are often connected to the processing unit 1220 through a user
10 input interface 1260 that is coupled to the system bus, but may be connected by
11 other interface and bus structures, such as a parallel port, game port, or a universal
12 serial bus (USB). A monitor 1291 or other type of display device is also
13 connected to the system bus 1221 via an interface, such as a video interface 1290.
14 In addition to the monitor 1291, computers may also include other peripheral
15 output devices such as speakers 1297 and printer 1296, which may be connected
16 through an output peripheral interface 1295.

17 The exemplary computer 1200 may operate in a networked environment
18 using logical connections to one or more remote computers, such as a remote
19 computer 1280. The remote computer 1280 may be a personal computer, a server,
20 a router, a network PC, a peer device or other common network node, and
21 typically includes many or all of the elements described above relative to
22 exemplary computer 1200, although only a memory storage device 1281 has been
23 illustrated in Fig. 12. The logical connections depicted in Fig. 12 include a local
24 area network (LAN) 1271 and a wide area network (WAN) 1273, but may also
25

1 include other networks. Such networking environments are commonplace in
2 offices, enterprise-wide computer networks, intranets, and the Internet.

3 When used in a LAN networking environment, the exemplary computer
4 1200 is connected to the LAN 1271 through a network interface or adapter 1270.
5 When used in a WAN networking environment, the exemplary computer 1200
6 typically includes a modem 1272 or other means for establishing communications
7 over the WAN 1273, such as the Internet. The modem 1272, which may be
8 internal or external, may be connected to the system bus 1221 via the user input
9 interface 1260, or other appropriate mechanism. In a networked environment,
10 program modules depicted relative to the exemplary computer 1200, or portions
11 thereof, may be stored in the remote memory storage device. By way of example,
12 and not limitation, Fig. 12 illustrates remote application programs 1285 as residing
13 on memory device 1281. It will be appreciated that the network connections
14 shown are exemplary and other means of establishing a communications link
15 between the computers may be used.

16 17 CONCLUSION

18 The foregoing describes an exemplary pane element. The subject matter
19 described above can be implemented in hardware, in software, or in both hardware
20 and software. In certain implementations, the exemplary pane element may be
21 described in the general context of computer-executable instructions, such as
22 program modules, being executed by a computer. Generally, program modules
23 include routines, programs, objects, components, data structures, etc. that perform
24 particular tasks or implement particular abstract data types. The subject matter can
25 also be practiced in distributed communications environments where tasks are

1 performed over wireless communication by remote processing devices that are
2 linked through a communications network. In a wireless network, program
3 modules may be located in both local and remote communications device storage
4 media including memory storage devices.
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25